



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|-------------|------------------------|---------------------|------------------|
| 10/560,839 | 06/08/2006 | Elmar Thurner | 2003P06167W0US | 6878 |
| 22116 7590 11/29/2009 SIEMENS CORPORATION INTELLECTUAL PROPERTY DEPARTMENT 170 WOOD AVENUE SOUTH ISELIN, NJ 08830 | | | | |
| EXAMINER GAMIL TEJAL | | | | |
| ART UNIT 2121 | | PAPER NUMBER | | |
| MAIL DATE 11/20/2009 | | DELIVERY MODE PAPER | | |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/560,839

Applicant(s)

THURNER, ELMAR

Examiner

TEJAL J. GAMI

Art Unit

2121

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 July 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 25, 27-31, 34-38, 40, 41 and 43-47 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 25, 27-31, 34-38, 40, 41 and 43-47 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 07 July 2009 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This office action is responsive to an AMENDMENT entered July 7, 2009 for the patent application 10/560839.

Status of Claims

2. Claims 25, 27-31, 34-38, 40, 41 and 43-47 were rejected in the last Office Action dated April 15, 2009.

As a response to the April 15, 2008 office action, Applicant has Amended claim 25 and 30.

Claims 25, 27-31, 34-38, 40, 41 and 43-47 are now presented for examination in this office action.

Drawings

3. Examiner thanks Applicant for submitting corrected drawings in response to the objection presented in the previous office action. The replacement drawings are accepted.

Claim Objections

4. Examiner thanks Applicant for amendments made to the claim in response to the claim objection presented in the previous office action. The claim objection has been withdrawn.

Claim Rejections - 35 USC § 112

5. Examiner thanks Applicant for amendments made to the claim in response to the 35 U.S.C. 112 second paragraph rejection made in the previous office action. The USC 112 rejection is thereby withdrawn.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 25, 27-31, 34-38, 40, 41 and 43-47 are rejected under 35 U.S.C. 102(b) as being anticipated by Stripf et al. (U.S. Patent Number: 6,263,487).

As to independent claim 25, Stripf discloses a method for executing a program for an industrial automation system (e.g., program for controlling and industrial process) (see Col. 1, Lines 11-15), comprising:

providing a computer unit (e.g., Figure 1) with:

input aids, output aids, a display device (e.g., operating and monitoring devices for displaying) (see Col. 3, Lines 23-33), modules and functions respectively representing sub-tasks of an automation solution (e.g., a class of software function blocks and a class of input/output modules) (see Col. 4, Lines

15-21), and a program which is structured from the modules and functions (e.g., control program formed by multiple software function blocks...the software function blocks have an object-oriented design) (see Col. 2, Lines 9-20; Figure 2 and Col. 2, Lines 44-46);

converting the modules and functions of the structured program into objects (e.g., translates the control program into an object-oriented machine language e.g., java byte code) (see Col. 2, Lines 44-55) to create a machine-independent program (e.g., the portability of the code ensures that a programmable controller with an execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40) in the form of a hierarchical tree (see byte code as a form of an hierarchical tree as further limited by applicant's claim 30; and see supporting document: wikipedia online encyclopedia, page 11 for an example of java byte code data structure such as trees); and

loading the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system (e.g., Java byte code loaded into programmable controllers) (see Col. 2, Lines 55-57), wherein the corresponding components of the automation system execute the machine-independent program present in the form of the at least one hierarchical tree with the aid of at least one object machine assigned to the

corresponding components of the automation system (e.g., the portability of the code ensures that a programmable controller with a execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40), and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the at least one hierarchical tree (e.g., operating and monitoring software program for creating and displaying a process image containing multiple image objects, where the image objects are in relation to in interaction with software function blocks of the control program) (see Col. 3, Lines 23-38); and

during or after loading of the machine-independent program, instantiating the operators using the at least one object machine into corresponding components of the automation system (e.g., software function blocks designed so that they are loadable and can be tied into the control program while it is running, the software function blocks have an object-oriented design) (see Col. 2, Lines 12-20); and

converting (e.g., translates) the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program (e.g., loaded into programmable controllers) in the form of an executable program or operator tree (e.g., compiler to create the machine codes) (see Col. 2, Line 44 to Col. 3, Line 3).

As to independent claim 38, Stripf discloses a device for executing a program for an industrial automation system (e.g., program for controlling and industrial process) (see Col. 1, Lines 11-15), comprising:

at least one computer unit with input aids, output aids and a display device (e.g., operating and monitoring devices for displaying) (see Col. 3, Lines 23-33);

a component for modeling and/or creating modules and functions, which respectively represent the sub-tasks of an automation solution (e.g., a class of software function blocks and a class of input/output modules) (see Col. 4, Lines 15-21);

a component for structuring the modules and functions and for networking the same, to form at least one hierarchical tree as at least one machine-independent program (e.g., software function blocks to be loadable directly into a programmable controller over the Internet and the Internet communications interface, so that they can be tied in to the control program while it is running, the software function blocks have an object-oriented design) (see Col. 1, Line 65 to Col. 2, Line 25; Figure 2 and Col. 2, Lines 44-46); and

a component to load the machine-independent program in the form of at least one hierarchical tree into the corresponding components of the automation system with the corresponding components of the automation system executing the machine-independent program present in the form of the at least one hierarchical tree (e.g., Java byte code loaded into programmable controllers) (see Col. 2, Lines 55-57), wherein at least one object machine is assigned to the corresponding components of the automation system to execute the machine-independent programs (e.g., the portability

of the code ensures that a programmable controller with a execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40), and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the hierarchical tree (e.g., operating and monitoring software program for creating and displaying a process image containing multiple image objects, where the image objects are in relation to in interaction with software function blocks of the control program) (see Col. 3, Lines 23-38);

a component to instantiate the operators using the at least one object machine during or after the loading of the machine-independent program into corresponding components of the automation system (e.g., software function blocks designed so that they are loadable and can be tied into the control program while it is running, the software function blocks have an object-oriented design) (see Col. 2, Lines 12-20); and

a component to convert (e.g., translates) the symbolic representation of the at least one hierarchical tree to physical addresses to generate a loadable program (e.g., loaded into programmable controllers) in the form of an executable program or operator tree (e.g., compiler to create the machine codes) (see Col. 2, Line 47 to Col. 3, Line 3).

As to independent claim 47, Stripf discloses a computer program implementing a method for executing a program for an industrial automation system (e.g., program for controlling and industrial process) (see Col. 1, Lines 11-15), comprising:

providing a computer unit with input aids, output aids and a display device (e.g., operating and monitoring devices for displaying) (see Col. 3, Lines 23-33), having modules and functions respectively representing sub-tasks of an automation solution being modeled and/or created using the input aids and optionally the display device (e.g., a class of software function blocks and a class of input/output modules) (see Col. 4, Lines 15-21), having the modules and functions being structured and networked using the input aids and optionally the display device as to form a hierarchical tree as a machine-independent program (e.g., software function blocks to be loadable directly into a programmable controller over the Internet and the Internet communications interface, so that they can be tied in to the control program while it is running, the software function blocks have an object-oriented design) (see Col. 1, Line 65 to Col. 2, Line 25; Figure 2 and Col. 2, Lines 44-46);

loading the machine-independent program in the form of the hierarchical tree into the corresponding components of the automation system (e.g., Java byte code loaded into programmable controllers) (see Col. 2, Lines 55-57), wherein the corresponding components of the automation system execute the machine-independent program present in the form of the hierarchical tree with the aid of at least one object machine assigned to the corresponding components of the automation system (e.g., the portability of the code ensures that a programmable controller with a execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40), and wherein the

at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the hierarchical tree (e.g., operating and monitoring software program for creating and displaying a process image containing multiple image objects, where the image objects are in relation to in interaction with software function blocks of the control program) (see Col. 3, Lines 23-38); and

during or after loading of the machine-independent program, instantiating the operators using the at least one object machine into corresponding components of the automation system (e.g., software function blocks designed so that they are loadable and can be tied into the control program while it is running, the software function blocks have an object-oriented design) (see Col. 2, Lines 12-20); and

converting (e.g., translates) the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program (e.g., loaded into programmable controllers) in the form of an executable program or operator tree (e.g., compiler to create the machine codes) (see Col. 2, Line 47 to Col. 3, Line 3).

As to dependent claim 27, Stripf teaches the method according to claim 25, wherein the machine-independent program is present in the form of at least one hierarchical object or operator tree in the corresponding components of the automation system and are processed interpretatively (e.g., byte code interpreter 10) (see Col. 2, Line 47 to Col. 3, Line 3).

As to dependent claim 28, Stripf teaches the method according to claim 27, wherein the machine-independent program is present in the form of at least one object

or operator tree with a structure equivalent or similar to the representation of the program in the display device (e.g., Figure 2) (see Col. 2, Lines 44-46).

As to dependent claim 29, Stripf teaches the method according to claim 25, wherein the machine-independent program is loaded into the corresponding components of the automation system using a machine-independent, symbolic representation of the hierarchical tree (e.g., Java byte code loaded into programmable controllers) (see Col. 2, Lines 55-57).

As to dependent claim 30, Stripf teaches the method according to claim 29, wherein the machine-independent and symbolic representation of the hierarchical tree is in the form of a byte code language or a markup language (e.g., byte code) (see Col. 2, Line 36).

As to dependent claim 31, Stripf teaches the method according to claim 25, wherein the object machine is configured as a real-time object machine with deterministic response and cycle times (e.g., cycle time) (see Col. 4, Lines 33-48).

As to dependent claim 34, Stripf teaches the method according to claim 25, wherein the object machine is implemented as a function unit that is closed and that processes the at least one hierarchical tree to a runtime system of the automated system (e.g., software function blocks to be loadable directly into a programmable controller over the Internet and the Internet communications interface, so that they can be tied in to the control program while it is running, the software function blocks have an object-oriented design) (see Col. 1, Line 65 to Col. 2, Line 25; Figure 2 and Col. 2, Lines 44-46).

As to dependent claim 35, Strif teaches the method according to claim 27, wherein the object machine is implemented in a distributed manner as at least one object, with the hierarchical object or operator tree processing itself (e.g., distributed programmable controller) (see Figure 1; and Col. 1, Lines 54-56).

As to dependent claim 36, Strif teaches the method according to claim 25, wherein the modules and functions are assigned model information and/or meta-information using the input aids and/or the display device (e.g., a class of software function blocks and a class of input/output modules) (see Col. 4, Lines 15-21).

As to dependent claim 37, Strif teaches the method according to claim 27, wherein the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects (e.g., the portability of the code ensures that a programmable controller with an execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40; Figure 2 and Col. 2, Lines 44-46).

As to dependent claim 40, Strif teaches the device according to claim 38, wherein the machine-independent program is present in the form of at least one object or operator tree with a structure equivalent or similar to the representation of the program in the display device (e.g., Figure 2) (see Col. 2, Lines 44-46).

As to dependent claim 41, Stripf teaches the device according to claim 38, wherein the object machine is configured as a real-time object machine with deterministic response and cycle times (e.g., cycle time) (see Col. 4, Lines 33-48).

As to dependent claim 43, Stripf teaches the device according to claim 38, further comprising a device for assigning model information and/or meta-information to the modules and functions (e.g., a class of software function blocks and a class of input/output modules) (see Col. 4, Lines 15-21).

As to dependent claim 44, Stripf teaches the device according to claim 38, wherein the object machine is implemented as a function unit that is closed and processes the at least one hierarchical tree to a runtime system of the automated invention (e.g., software function blocks to be loadable directly into a programmable controller over the Internet and the Internet communications interface, so that they can be tied in to the control program while it is running, the software function blocks have an object-oriented design) (see Col. 1, Line 65 to Col. 2, Line 25; Figure 2 and Col. 2, Lines 44-46).

As to dependent claim 45, Stripf teaches the device according to claim 38, wherein the object machine is implemented in a distributed manner as at least one object, with the hierarchical object or operator tree processing itself (e.g., distributed programmable controller) (see Figure 1; and Col. 1, Lines 54-56).

As to dependent claim 46, Stripf teaches the device according to claim 38, wherein the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or

infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or infrastructure function can be used by all the objects (e.g., the portability of the code ensures that a programmable controller with an execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller) (see Col. 2, Lines 35-40; Figure 2 and Col. 2, Lines 44-46).

Response to Arguments

8. Applicant's amendment and arguments filed July 7, 2009 have been fully considered. The amendment does not overcome the original art rejection and the arguments are not persuasive. The following are the Examiner's observations in regard thereto.

Applicant Argues:

Stripf does not expressly or inherently teach or suggest "a machine-independent program in the form of a hierarchical tree".

Examiner Responds:

Examiner is not persuaded. See prior art Col. 2, Lines 35-40 for byte code (e.g., hierarchical tree, as further limited by applicant's claim 30). And see supporting document: wikipedia online encyclopedia, page 11 for an example of java byte code data structure such as trees. Under such considerations, the prior art anticipates the claim as written, a machine-independent program in the form of a hierarchical tree.

Applicant Argues:

Nothing in Strip expressly or inherently describes "containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects" as claimed in dependent claims 37 and 46.

Examiner Responds:

Examiner is not persuaded. See supporting documents: wikipedia online encyclopedia, for an explanation of a container being a class; and see prior art Col. 4, Lines 15-30 for a class of software function blocks. Thereby supporting the anticipation of the claims as written.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tejal J. Gami whose telephone number is (571) 270-1035. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert DeCady can be reached on (571) 272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Albert DeCady/
Supervisory Patent Examiner, Art
Unit 2121

/TJG/